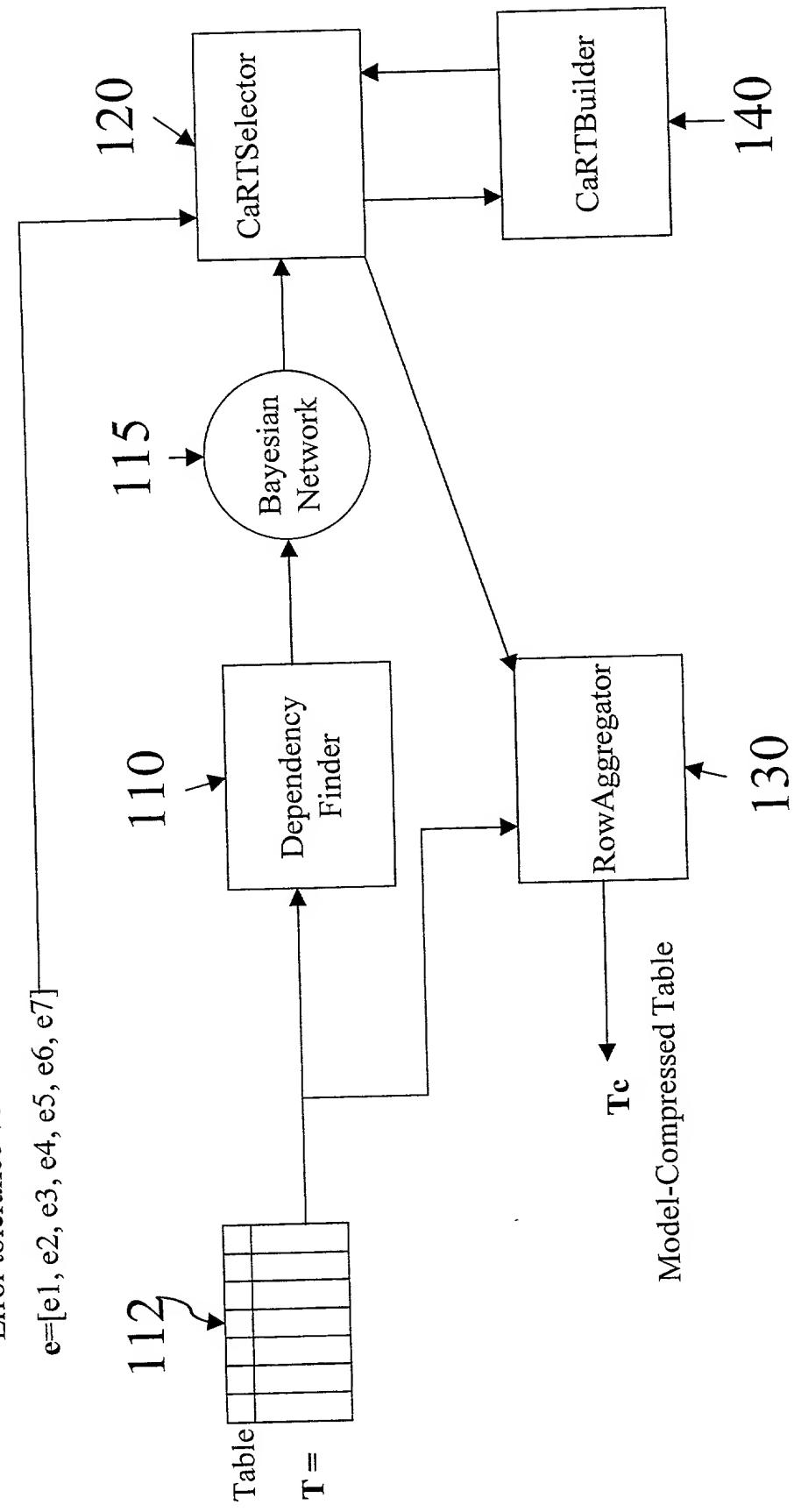


123  
Error tolerance vector  
e=[e1, e2, e3, e4, e5, e6, e7]



100 ↗

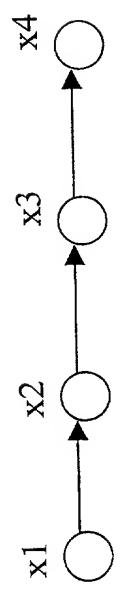
FIGURE 1

```

procedure Greedy (T(X),  $\bar{e}$ , G,  $\theta$ )
Input:   n-attribute table T and n-vector of error tolerances  $\bar{e}$ ;
         Bayesian network G on the set of attributes X and
         threshold  $\theta$  on the relative benefit for selecting a
         CaRT predictor.
Output:  A set of materialized (predicted) attributes  $X_{mat}$  ( $X_{pred}$ 
         =  $X - X_{mat}$ ) and a CaRT predictor for each  $X_i \in X_{pred}$ .
begin
1.    $X_{mat} := X_{pred} := \emptyset$ 
2.   let  $\langle X_1, X_2, \dots, X_n \rangle$  be the attributes in X sorted in
     topological order of G
3.   for  $i := 1, \dots, n$ 
4.   if  $\pi(X_i) = \emptyset$  then  $X_{mat} := X_{mat} \cup \{X_i\}$  /*  $X_i$  must be
     materialized if it has no parents in G */
5.   else
6.     M := BuildCaRT ( $X_{mat} \rightarrow X_i, e_i$ )
7.     if (MaterCost ( $X_i$ ) / PredCost ( $X_{mat} \rightarrow X_i$ ) >  $\theta$ ) then  $X_{pred} :=$ 
       $X_{pred} \cup \{X_i\}$ 
8.   else  $X_{mat} := X_{mat} \cup \{X_i\}$ 
9.   end
10.  end
end

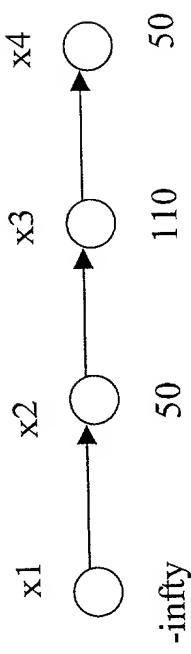
```

FIGURE 2: The Greedy CaRT Selection Algorithm



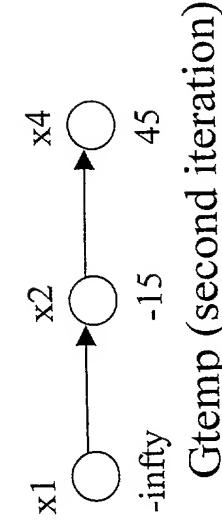
Bayesian Network G

FIGURE 3A



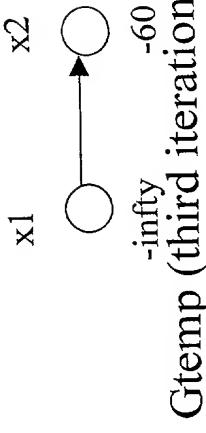
(b) Gtemp (first iteration)

FIGURE 3B



Gtemp (second iteration)

FIGURE 3C



FIGURES 3A-3D

FIGURE 3D

```

procedure MaxIndependentSet (T(X),  $\bar{e}$ , G, neighborhood() )
Input:   n-attribute table T and n-vector of error tolerances  $\bar{e}$ ;
         Bayesian network G on the set of attributes X and function
         neighborhood() defining the "predictive neighborhood" of a
         node  $X_i$  in G (e.g.,  $\pi(X_i)$  or  $\beta(X_i)$ ).
Output:  A set of materialized (predicted) attributes  $X_{mat}$  ( $X_{pred} = X - X_{mat}$ ) and a CaRT predictor  $PRED(X_i) \rightarrow X_i$  for each  $X_i \in X_{pred}$ .
begin
1.    $X_{mat} := X$ ,  $X_{pred} := \emptyset$ 
2.    $PRED(X_i) := \emptyset$  for all  $X_i \in X$ ,  $improve := true$ 
3.   while ( $improve \neq false$ ) do
4.       for each  $X_i \in X_{mat}$ 
5.            $mater\_neighbors(X_i) :=$ 
              $(X_{mat} \cap neighborhood(X_i)) \cup \{PRED(X) : X \in neighborhood$ 
              $(X_i), X \in X_{pred}\} - \{X_i\}$ 
6.            $M := BuildCaRT(Mater\_neighbors(X_i) \rightarrow X_i, e_i)$ 
7.           let  $PRED(X_i) \subseteq mater\_neighbors(X_i)$  be the set of
             predictor attributes used in M
8.            $cost\_change_i := 0$ 
9.           for each  $X_j \in X_{pred}$  such that  $X_i \in PRED(X_j)$ 
10.           $NEW\_PRED_i(X_j) := PRED(X_j) - \{X_i\} \cup PRED(X_i)$ 
11.           $M := BuildCaRT(NEW\_PRED_i(X_j) \rightarrow X_j, e_j)$ 
12.          set  $NEW\_PRED_i(X_j)$  to the (sub)set of
             predictor attributes used in M
13.           $cost\_change_i := cost\_change_i + (PredCost(PRED$ 
              $(X_j) \rightarrow X_j) - PredCost(NEW\_PRED_i(X_j) \rightarrow X_j))$ 
14.      end
15.  end
16.  build an undirected, node-weighted graph  $G_{temp} = (X_{mat},$ 
    $E_{temp})$  on the current set of materialized
   attributes  $X_{mat}$ , where:
17.  (a)  $E_{temp} := \{(X, Y) : \text{for all pairs } X, Y \in X_{pred}\} \cup$ 
       $\{(X_i, Y) : \text{for all } Y \in X_{mat}\}$ 
18.  (b)  $weight(X_i) := MaterCost(X_i) - PredCost(PRED(X_i)$ 
       $\rightarrow X_i) + cost\_change_i$  for each  $X_i \in X_{mat}$ 
21.   $S := FindWMIS(G_{temp})$  /* select (approximate) maximum
   weight independent set in  $G_{temp}$ 
22.                               as "maximum-benefit" subset of
   predicted attributes */
23.  if ( $\sum_{X \in S} weight(X) \leq 0$ ) then  $improve := false$ 
24.  else /* update  $X_{mat}$ ,  $X_{pred}$ , and the chosen CaRT predictors */
25.      for each  $X_j \in X_{pred}$ 
26.          if ( $PRED(X_j) \cap S = \{X_i\}$ ) then  $PRED(X_j) :=$ 
              $NEW\_PRED_i(X_j)$ 
27.      end
28.       $X_{mat} := X_{mat} - S$ ,  $X_{pred} := X_{pred} \cup S$ 
29.  end
30. end /* while */
end

```

FIGURE 4: The MaxIndependentSet CaRT Selection Algorithm

```

procedure LowerBound (N,  $e_i$ , b)
Input: Leaf N for which lower bound on subtree cost is to be
       computed; error tolerance  $e_i$  for attribute  $X_i$ ; bound b
       on the maximum number of internal nodes in subtree
       rooted at N.
Output: Lower bound  $L(N)$  on cost of subtree rooted at N.
begin
1.   for i := to r
2.       minOut [i,0] :=i
3.   for J := 1 to b + 1
4.       minOut [0,j] :=0
5.   l :=0
6.   for i := 1 to r
7.       while  $x_i - x_{i+1} > 2e_i$ 
8.           l := l + 1
9.   for j := 1 to b + 1
10.      minOut [i,j] := min {minOut [i - 1,j] + 1, minOut [l,j-1]}
11. end
12.  $L(N) := \infty$ 
13. for J := 0 to b
14.      $L(N) := \min \{L(N), 2j + 1 + j \log (|X_i|) + (j + 1 + \min_{(r,j+1)} \log (|\text{dom}(X_i)|))\}$ 
15.  $L(N) := \min \{L(N), 2b + 3 + (b + 1) \log (|X_i|) + (b + 2) \log (|\text{dom}(X_i)|)\}$ 
16. return  $L(N)$ 
end

```

FIGURE 5: Algorithm for Estimating Lower Bound on Subtree Cost

FIGURE 6

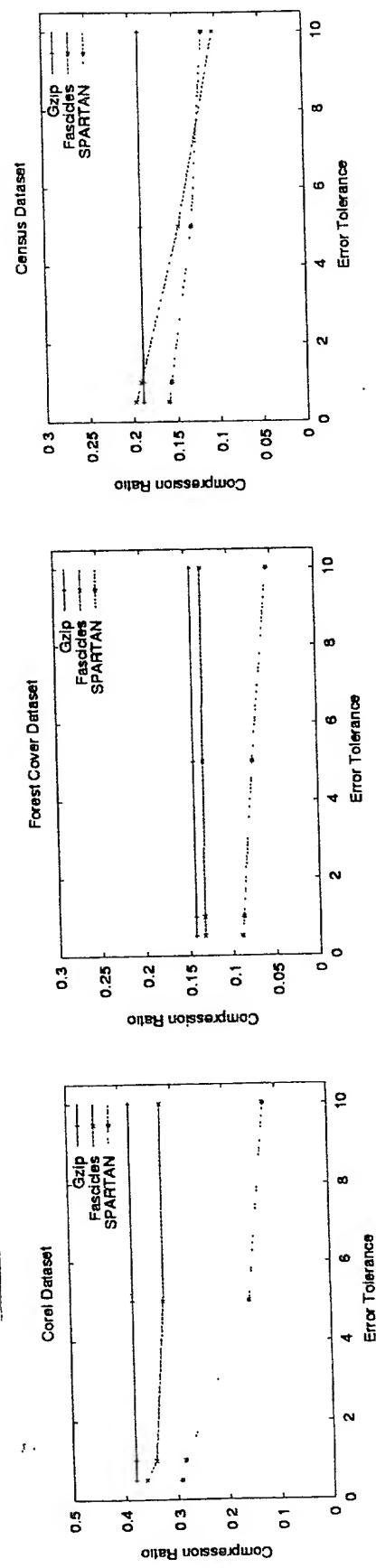


FIGURE 7.

